

Penerapan Metode Steepest Ascent Hill Climb pada Permainan Puzzle

Hairul Anam*, Feby Sabilhul Hanafi**, Ahmad Fauzal Adifia***,
Ahmad Firdaus Ababil****, Saiful Bukhori*****

* Student at University of Jember, ** Student at University of Jember,

*** Student at University of Jember, **** Student at University of Jember,

***** Lecture at University of Jember

* hairulanam21@gmail.com, ** febysabilhul@gmail.com,

*** adifia19@gmail.com, **** ahmad.firdaus.1054@gmail.com,

***** saiful.ilkom@unej.ac.id

ABSTRAK

Puzzle merupakan salah satu contoh penerapan dari kecerdasan buatan, dalam proses penyelesaiannya banyak terdapat algoritma-algoritma pencarian yang dapat diterapkan. Solusi 8 *puzzle* akan lebih cepat diperoleh jika digunakan prinsip *array* dengan variasi algoritma *Steepest-Ascent Hill Climbing (Hill Climbing)* dengan memilih kemiringan yang paling tajam / curam) dengan parameter heuristik posisi yang benar dan heuristik jarak serta dikombinasikan dengan *LogList* sebagai penyimpanan *state-state* yang pernah dilalui untuk menanggulangi permasalahan pada algoritma *hill climbing* itu sendiri dan terhindar dari *looping state* yang pernah dilalui. *Steepest Ascent Hill Climbing* merupakan metode algoritma yang banyak digunakan untuk permasalahan optimasi. Penerapan Algoritma SAHC (*Steepest Ascent Hill Climbing*) pada *puzzle* dibutuhkan agar permainan selesai dengan waktu yang optimal.

Kata Kunci: *Puzzle, Steepest Ascent, Hill Climb, Permainan, Kecerdasan Buatan*

1. Pendahuluan

Salah satu permainan yang umum dimainkan yaitu *puzzle*, banyak jenis permainan *puzzle* yang beredar di masyarakat baik yang masih tradisional bahkan sampai permainan yang berbasis *android* atau *smartphone* dengan beragam jenis dan permainan yang tidak hanya diperuntukkan bagi kalangan anak semata namun semua kalangan bisa menikmatinya. Menurut Patmonodewo (Misbach, Muzamil, 2010) kata *puzzle* berasal dari Bahasa Inggris yang berarti teka-teki atau bongkar pasang, media *puzzle* merupakan media sederhana yang dimainkan dengan bongkar pasang [3]. Dari pengertian tersebut dapat disimpulkan bahwa *puzzle* merupakan permainan yang edukatif, sangat berguna bagi perkembangan otak terutama bagi anak dalam proses pematangan berfikir. *Puzzle* sendiri dapat digunakan sebagai media pembelajaran. Sebagai media bermain dan belajar, media *puzzle* paling umum digunakan dalam menerapkan pembelajaran kepada anak-anak sekolah karena sederhana dan mencakup pembelajaran emosional dan mental yang diperlukan bagi anak seperti sikap tenang, sabar, dan ketekunan dalam menyelesaikan masalah, yang mana semua itu sangat diperlukan dan jika dilakukan dengan konsisten akan membentuk pola kepribadian yang baik bagi anak kedepannya.

Terdapat beberapa metode untuk membuat sebuah game *puzzle* tersebut diantaranya metode *Steepest Ascent Hill Climbing*, suatu metode pencarian berdasarkan nilai heuristik suatu fungsi yang memberikan nilai untuk perkiraan suatu solusi, metode ini sering digunakan untuk permasalahan optimasi. Pada metode *Steepest Ascent Hill Climbing* yang diimplementasikan pada permainan *puzzle* proses pengujian dilakukan dengan menggunakan fungsi heuristik. Pembangkitan keadaan berikutnya sangat tergantung pada *feedback* dari prosedur pengesanan “Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin”.

Langkah penyelesaian optimal untuk menyelesaikan *puzzle*, pada penelitian ini menggunakan metode *Steepest Ascent Hill Climbing*. Langkah-langkah dalam perhitungan SAHC (*Steepest Ascent Hill Climbing*) yaitu : (1) menghitung kotak yang menempati tempat yang benar, (2) hitung pergerakan yang memungkinkan. (3) mendapatkan nilai $h(n)$ menggunakan perhitungan manual dengan menggunakan penjumlahan kotak yang menempati tempat yang benar, (4) membandingkan nilai heuristik dari pergerakan yang mungkin, (5) menerapkan alur pencarian algoritma SAHC (*Steepest Ascent Hill Climbing*) dengan nilai heuristik $h(n)$ yang telah diperoleh [2].

2. Metode Steepest Ascent Hill Climbing

Steepest Ascent Hill Climbing merupakan metode algoritma yang banyak digunakan untuk permasalahan optimasi. Salah satu penerapannya adalah untuk mencari rute yang terpendek dengan cara memaksimalkan atau meminimumkan nilai dari fungsi optimasi yang ada. Secara harfiah *steepest* berarti paling tinggi, sedangkan *ascent* berarti kenaikan. Dengan demikian *steepest ascent* berarti kenaikan paling tinggi. Jadi prinsip dasar dari metode ini adalah mencari kenaikan paling tinggi dari keadaan sekitar untuk mencapai nilai yang paling optimal [1].

Algoritma ini memeriksa titik (node), yaitu biaya (*cost*) yang dibutuhkan untuk mencapai sebuah node dan heuristik $h(n)$ yaitu *cost* yang didapat dari node ke tujuan. Sehingga dapat dirumuskan sebagai berikut [2] :

$$f(n) = h(n)$$

- $f(n)$ = perkiraan total cost terendah dari setiap *path* yang akan dilalui dari node n ke node tujuan.
- $h(n)$ = perkiraan heuristik atau *cost* atau *path* dari node n ke tujuan.

Untuk menentukan nilai $h(n)$ ditunjukkan oleh persamaan 1, berikut :

$$1. h(n) =$$

dimana :

$h(n)$ = nilai heuristik untuk node / titik n

X_n = nilai koordinat X dari node / titik n

Y_n = nilai koordinat Y dari node / titik n

X_{goal} = nilai koordinat X dari node / titik tujuan

Y_{goal} = nilai koordinat Y dari node / titik tujuan

Metode *steepest ascent hill climbing* ini merupakan pengembangan dari metode *simple hill climbing*. Bedanya adalah *simple hill climbing* menentukan *next state* dengan membandingkan *current state* (titik awal) dengan satu successor (titik persimpangan) dan successor pertama yang lebih baik akan dipilih menjadi *next state*. Sedangkan *steepest ascent* akan membandingkan *current state* dengan semua *successor* yang ada didekatnya sehingga dalam *steepest ascent hill climbing*, *next statenya* merupakan *successor* yang paling baik atau paling mendekati tujuan [5].

Berikut adalah perbandingan algoritma *steepest ascent hill climbing* dengan *hill climbing* [2]:

Algoritma *simple hill climbing*:

1. Evaluasi *state* awal, jika *state* awal sama dengan tujuan, maka proses berhenti. Jika tidak sama dengan tujuan maka lanjutkan proses dengan membuat *state* awal sebagai *state* sekarang.
2. Kerjakan langkah berikut sampai solusi ditemukan atau sampai tidak ada lagi operator baru yang dapat digunakan dalam *state* sekarang :
 - a. Cari sebuah operator yang belum pernah digunakan dalam *state* sekarang dan gunakan operator tersebut untuk membentuk *state* baru.
 - b. Evaluasi *state* baru.
 - i. Jika *state* baru adalah tujuan, maka proses berhenti
 - ii. Jika *state* baru tersebut bukan tujuan tetapi *state* baru lebih baik dari pada *state* sekarang, maka buat *state* baru menjadi *state* sekarang.
 - iii. Jika *state* baru tidak lebih baik dari pada *state* sekarang, maka lanjutkan ke langkah

Algoritma *steepest ascent hill climbing*

1. Evaluasi keadaan awal (*Initial State*). Jika keadaan awal sama dengan tujuan (*Goal state*) maka kembali pada *initial state* dan berhenti berproses. Jika tidak maka *initial state* tersebut dijadikan sebagai *current state*.
2. Mulai dengan *current state = initial state*.
3. Dapatkan semua pewaris (*successor*) yang dapat dijadikan *next state* pada *current statenya* dan evaluasi *successor* tersebut dengan fungsi evaluasi dan beri nilai pada setiap *successor* tersebut. Jika salah satu dari *successor* tersebut mempunyai nilai yang lebih baik dari *current state* maka jadikan *successor* dengan nilai yang paling baik tersebut sebagai *new current state*. Lakukan

operasi ini terus menerus hingga tercapai *current state* = *goal state* atau tidak ada perubahan pada *current statenya*.

Metode *Steepest Ascent Hill Climbing*, menggunakan algoritma sebagai berikut ini [2]:

1. Mulai dari keadaan awal, lakukan pengujian, jika merupakan tujuan maka berhenti, dan jika tidak lanjutkan dengan keadaan sekarang sebagai keadaan awal.
 2. Kerjakan hingga tujuan tercapai atau hingga iterasi tidak memberikan perubahan pada keadaan sekarang.
 - a. Misalkan SUCC adalah suatu *state* yang menjadi suksesor dari *current state*.
 - b. Untuk setiap operator bisa dilakukan terhadap *current state*, kerjakan:
 - i. Aplikasi operator tersebut dan bangkitkan *new state*.
 - ii. Evaluasi *new state*. Jika merupakan *goal state*, kembalikan *state* ini sebagai solusi dan keluar dari program. Jika bukan *goal state*, bandingkan *new state* dengan SUCC. Jika *new state* lebih baik dari SUCC, maka ganti SUCC dengan *new state*. Jika tidak lebih baik SUCC tidak perlu diganti.
 - c. Jika SUCC lebih baik dari *current state*, maka ganti *current state* dengan SUCC.
- Pada *Steepest Ascent Hill Climbing*, ada tiga masalah yang mungkin terjadi, yaitu [4]:
- a. *Local Optimum* : keadaan semua tetangga lebih buruk atau sama dengan keadaan dirinya.
 - b. *Plateau* : keadaan semua tetangga sama dengan keadaan dirinya.
 - c. *Ridge* : local optimum yang lebih disebabkan karena ketidak mampuan untuk menggunakan dua operator sekaligus.

3. Studi Kasus

Skema pengujian kasus yang dilakukan yaitu pada *puzzle* yang terdiri dari 8 angka yang harus diurutkan sehingga mencapai posisi yang diinginkan. Tujuan pengujian ini adalah menyelesaikan permainan *puzzle* dengan menggunakan metode *steepest ascent hill climbing*, yaitu memperhitungkan nilai heuristik yang dimiliki oleh setiap slide *puzzle*. Dimana pada *puzzle* diberikan *initial state* seperti gambar 2.

1	2	3
7	8	4
6		5

Gambar 2. Posisi awal puzzle (Initial state dengan heuristic 5)

Disini diketahui bahwa banyaknya angka yang menempati benar adalah 5 sedangkan 3 angka sisanya masih berada di posisi yang salah, maka banyaknya pilihan langkah yang bisa diambil dari posisi tersebut sebanyak 3 langkah, yaitu atas, kanan dan kiri.

1	2	3
8		4
7	6	5

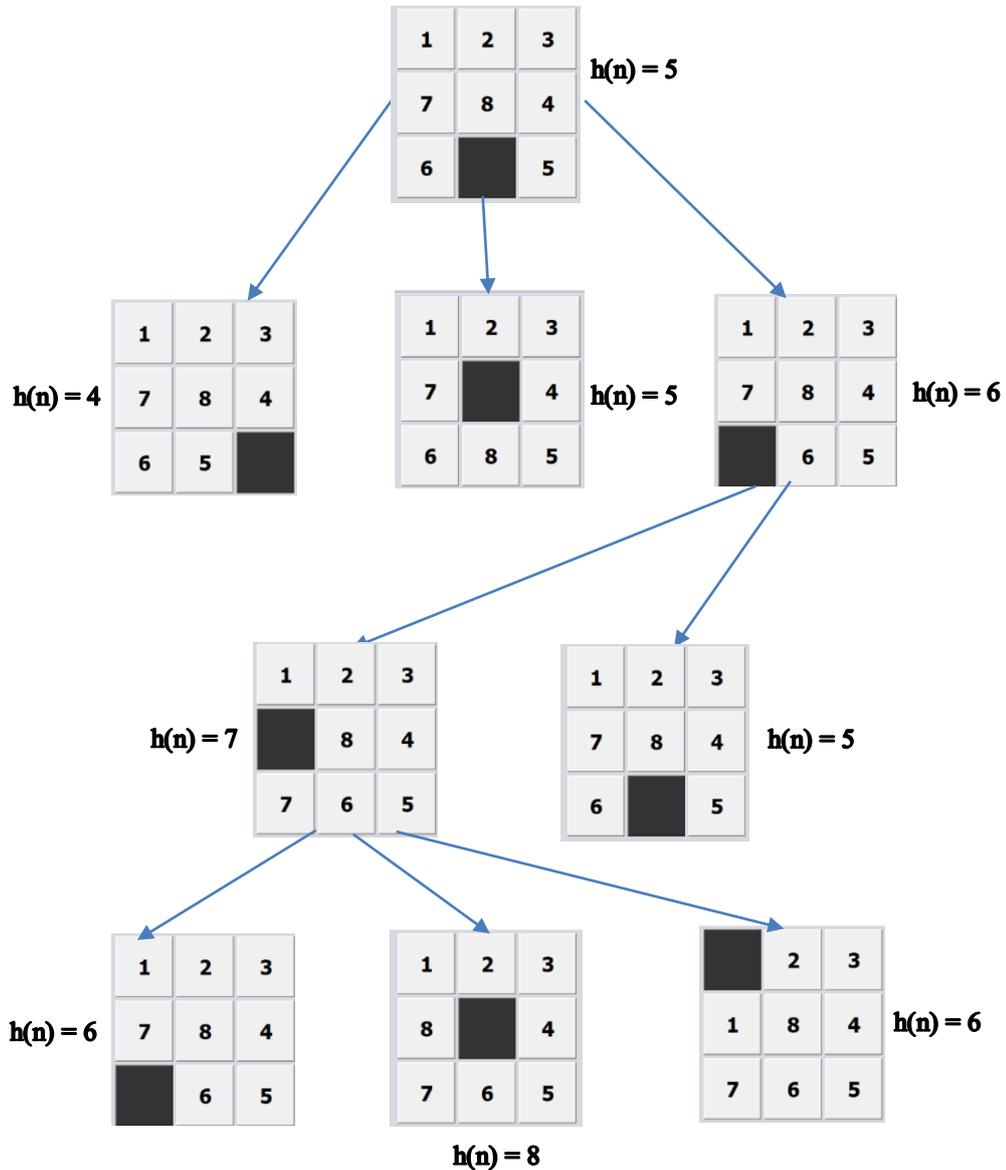
Gambar 3. Posisi akhir Puzzle

(Goal state dengan heuristic 8)

Sedangkan posisi akhir (*goal*) dari permainan ini yaitu mengurutkan semua angka sesuai Gambar 3 dimana posisi angka dari 1 sampai 8 melingkari kotak perpindahan (kotak hitam).

Penyelesaian:

Berikut adalah langkah dari penyelesaian *puzzle* tersebut dengan *steepest ascent hill climbing*:



Gambar 4. Alur penyelesaian puzzle menggunakan Steept Accent

Dari Gambar 4, setiap *slide puzzle* kita ibaratkan sebagai node atau titik yang akan dilalui untuk mencapai *goal state*. Dimana pada setiap node memiliki nilai heuristik yang diperoleh dari jumlah angka yang menempati posisi benar.

Iterasi 1 :

Pada iterasi awal, terdapat 3 kemungkinan yang dapat dilakukan yaitu kemungkinan pertama menggeser angka 5 ke kiri dengan diperoleh hasil bahwa angka yang menempati posisi benar yaitu 1, 2, 3, 4 dan angka menempati posisi salah yaitu 5, 6, 7, dan 8. Nilai heuristik diperoleh berdasarkan jumlah angka yang

menempati posisi benar kemudian kita beri nilai 1 sedangkan angka yang menempati posisi salah kita beri nilai 0 sehingga :

$$\begin{aligned} \text{Angka} &= 1-2-3-4-5-6-7-8 \\ \text{Nilai per angka} &= 1+1+1+1+0+0+0+0 \\ \text{Nilai heuristik} &= 4 \end{aligned}$$

Kemungkinan kedua dari iterasi pertama yaitu menggeser angka 8 kebawah. Dengan pemberian nilai sesuai kemungkinan pertama (yaitu 1 untuk angka pada posisi benar dan 0 jika salah), maka diperoleh hasil :

$$\begin{aligned} \text{Angka} &= 1-2-3-4-5-6-7-8 \\ \text{Nilai per angka} &= 1+1+1+1+1+0+0+0 \\ \text{Nilai heuristik} &= 5 \end{aligned}$$

Kemungkinan ketiga dari iterasi pertama diperoleh hasil :

$$\begin{aligned} \text{Angka} &= 1-2-3-4-5-6-7-8 \\ \text{Nilai per angka} &= 1+1+1+1+1+1+0+0 \\ \text{Nilai heuristik} &= 6 \end{aligned}$$

Dari iterasi 1 tersebut hasil yang diambil adalah kemungkinan ketiga, karena pada kemungkinan tersebut mempunyai nilai heuristik yang paling besar diantara ketiga kemungkinan yang dapat dilakukan.

Untuk iterasi 2 dan seterusnya, langkah pengerjaannya menggunakan perhitungan yang sama dengan iterasi 1. Dalam kasus ini goal sudah tercapai dengan 3 iterasi. Berikut adalah hasil dari setiap iterasi:

Tabel 1. Hasil Iterasi Pemecahan Puzzle 8 angka

Iterasi	Nilai per angka (1-2-3-4-5-6-7-8)	Heuristik
	1+1+1+1+1+0+0+0	5
1	1+1+1+1+1+1+0+0	6
2	1+1+1+1+1+1+1+0	7
3	1+1+1+1+1+1+1+1	8

Dari tabel diatas, dapat dilihat bahwa nilai heuristik terus bertambah hingga mencapai nilai maksimum sesuai dengan jumlah angka pada *puzzle* yaitu 8. Itu artinya, pada kasus ini metode *steepest ascent* mencapai *global optimum* dan tidak terjebak pada *local optimum*.

4. Kesimpulan

Dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa metode *steepest ascent hill climbing* bisa diterapkan dalam permainan *puzzle* dengan mempertimbangkan nilai heuristik dari langkah selanjutnya. Penyelesaian *puzzle* dengan *steepest ascent* bergantung pada inialisasi *state* awal yang diberikan karena hal itu akan berpengaruh pada nilai heuristik yang dimiliki. Oleh karenanya adakalanya *puzzle* mencapai posisi terbaik yang diinginkan bisa disebut *global optimum*. Namun terkadang akan mencapai *local optimum* yaitu ketika posisi *puzzle* tidak mencapai *goal state*.

Daftar Pustaka

- [1] Kusumadewi, S. (2003). *Artificial Intelligence*. Yogyakarta: Graha Ilmu
- [2] Zemma, L. A., & Qur, A. (2015). *Penerapan Metode Steepest Ascent Hill Climbing Pada Model Pencarian Rute Terdekat Fasilitas Pelayanan Darurat Di Kota Bogor Berbasis Android*.
- [3] Puzzle (online) (<http://www.academia.edu/9717051/>, diakses pada tanggal 19 Oktober 2017).
- [4] Zakiah, A., & Masalah, R. (2012). *Penyelesaian Masalah 8 Puzzle Dengan Algoritma Hill Climbing*, (Sentika), 158–163.
- [5] Uriawan, W., Faroqi, A., Fathonah, R., Informatika, J. T., Sains, F., & Negeri, U. I. (2015). Game slider puzzle. *Pembuatan Game Slider Puzzle Menggunakan Metode Steepest Ascent Hill Climbing Berbasis Android*, IX(1), 204–221.